impara l'elettronica

digitale

...e costruisci il tuo LABORATORIO DIGITALE

6,90 €

15

9 771824 363008 41015

Peruzzo & C.

IN REGALO in questo fascicolo

1 Circuito integrato 4093
2 Condensatori elettrolitici da 10 μF
2 LED rossi

IN REGALO nel prossimo fascicolo

1 Cavetto a quattro fili con due connettori femmina a quattro contatti
1 Cavetto a quattro fili con un connettore femmina a quattro contatti

COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartellette, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede.
Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail:
elettronicadigitale@microrobots.it

Hardware Montaggio e prove del laboratorio

Digitale di base Esercizi con i circuiti digitali

Digitale avanzato Esercizi con i circuiti sequenziali

Microcontroller Esercizi con i microcontroller

Direttore responsabile:
ALBERTO PERUZZO
Direttore Grandi Opere:
GIORGIO VERCELLINI
Consulenza tecnica e traduzioni:
CONSULCOMP S.n.c.
Pianificazione tecnica
LEONARDO PITTON

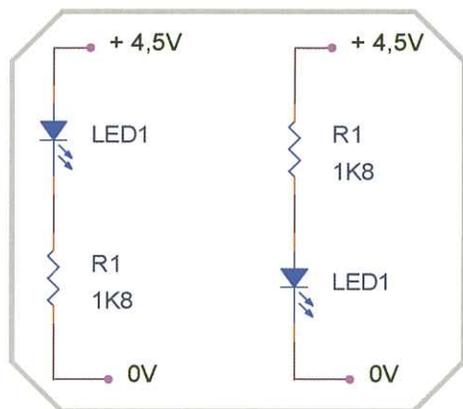
Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (Mi). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Staroffset s.r.l., Cernusco S/N (MI). Distribuzione SO.DI.P. S.p.A., Cinisello Balsamo (MI).

"ELETTRONICA DIGITALE"
si compone di
70 fascicoli settimanali
da suddividere
in 2 raccoglitori.

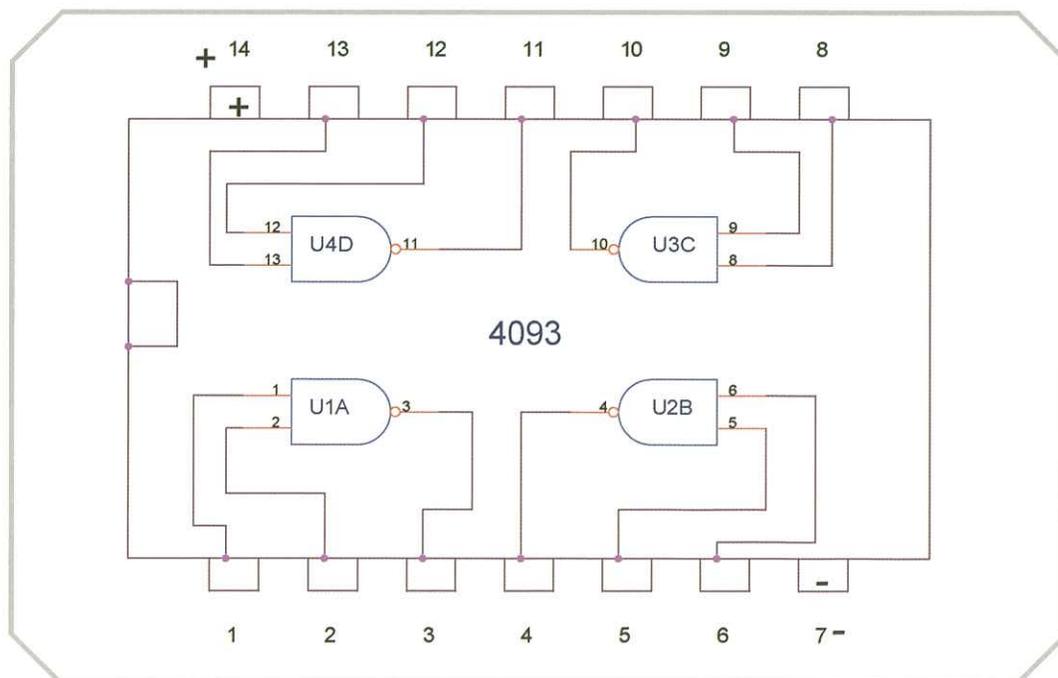
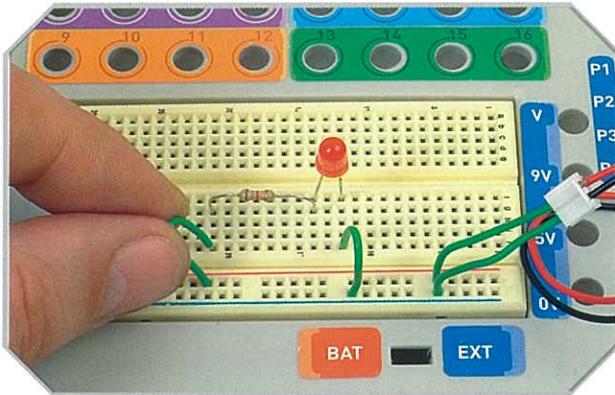
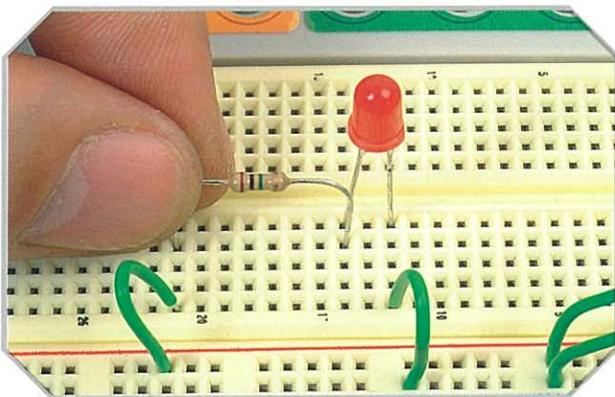
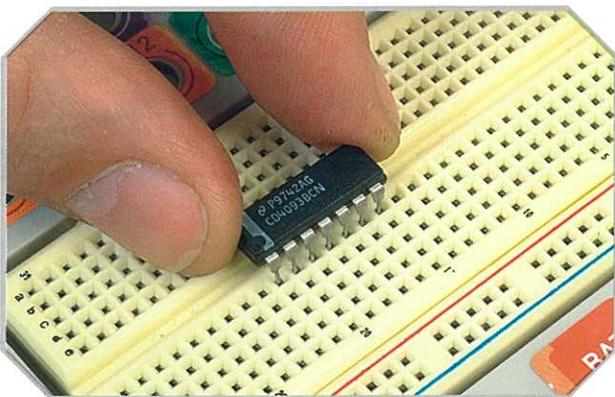
RICHIESTA DI NUMERI ARRETRATI. Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (MI). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammonteranno a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. IMPORTANTE: è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

Componenti e loro utilizzo

Componenti di questo fascicolo.

Continuiamo a fornire componenti per realizzare esperimenti sulla scheda Bread Board. Vi viene fornito un circuito integrato 4093 che contiene 4 porte NAND da due ingressi Trigger Schmitt, due condensatori elettrolitici da 10 µF di capacità e due diodi LED da 5 mm di colore rosso. Con questi componenti e il materiale fornito finora è già possibile montare alcuni circuiti sperimentali e verificarne il funzionamento.

Il LED ha bisogno di una resistenza che limiti la corrente.

4093

Questo circuito integrato contiene al suo interno quattro porte NAND a due ingressi. Nei circuiti digitali, per ragioni di chiarezza grafica e per eliminare le linee, spesso negli schemi si omettono i terminali corrispondenti all'alimentazione. Il terminale positivo dell'alimentazione è il 14 e il negativo il 7. Le quattro porte sono indipendenti. Ciò che di solito si indica è la numerazione di ogni terminale utilizzato.

Schema interno del circuito integrato 4093.

Prova di collegamento del LED con una resistenza da 1K8.

Se la resistenza è molto alta il LED non si illumina. In questo caso la resistenza è da 1 M.

Prima di inserire l'integrato bisogna allineare perfettamente i terminali ai fori della scheda.

I LED

I diodi elettroluminescenti, meglio noti come LED, emettono luce quando sono attraversati da una corrente diretta compresa normalmente tra 1 e 10 mA. Sono dispositivi direzionali e come tutti i diodi lasciano circolare la corrente in un solo verso, poiché quella che passa in senso inverso è praticamente trascurabile.

Il LED è un indicatore luminoso a basso consumo e con un'alta resistenza meccanica se paragonato a una qualsiasi lampada a incandescenza, è assai utile per rilevare livelli di tensione e verificare il corretto funzionamento di molti circuiti. Tuttavia occorre tener presente che il punto del circuito dove viene collegato deve poter fornire corrente sufficiente per provocare l'illuminazione del LED.

Condensatori elettrolitici

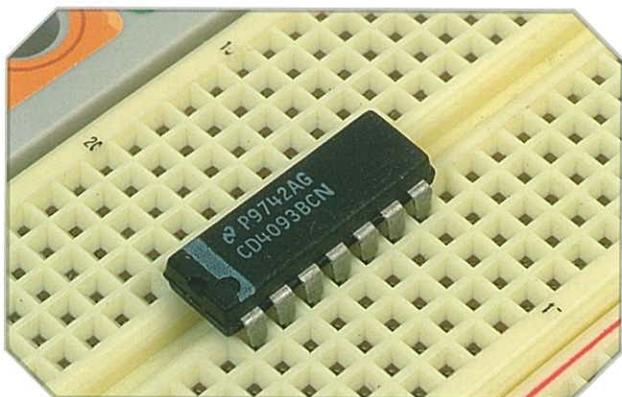
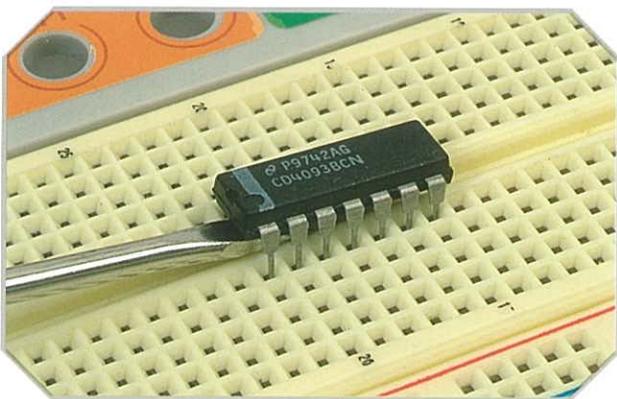
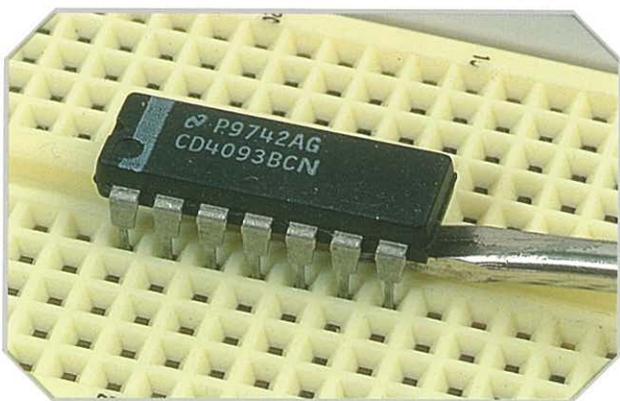
Questo tipo di condensatore ha il vantaggio di fornire una grande capacità in un volume minimo, però ha lo svantaggio di avere polarità, ovvero ha un polo positivo e uno negativo.

Il terminale positivo è solitamente più lungo e non deve essere tagliato per evitare errori, anche se sul contenitore del componente è indicato uno dei poli, il + o il -, vicino al terminale corrispondente per poter identificare sempre e in modo sicuro la polarità. Bisogna evitare gli errori, che possono danneggiare o, addirittura, far esplodere il componente se si collega in modo inverso, specialmente se utilizzato nelle alimentazioni.

Limitazione di corrente

È molto importante ricordare che il LED non limita la corrente che circola attraverso lo stesso, è necessario collegare in serie una resistenza che limiti la corrente, in caso contrario il LED si potrebbe distruggere a causa del passaggio di una corrente eccessiva.

In fase di progetto bisognerebbe tenere in considerazione l'illuminazione ambiente in modo da utilizzare la minima corrente per alimentare il LED; questo comporta un importante risparmio di energia, che potrebbe risultare importante se il dispositivo è alimentato a batterie.

È facile testare il LED e fare alcune prove

Per fare in modo che i piedini abbiano un buon contatto bisogna inserire a fondo l'integrato.

Per togliere l'integrato utilizzeremo un cacciavite per estrarlo da un lato.

Non bisogna tirare l'integrato fino a quando non sia estratto da entrambi i lati.

utilizzando la scheda Bread Board e, a questo scopo, sarà necessario mettere in serie allo stesso LED una resistenza, ad esempio da 1K8.

Il positivo del LED, ovvero l'anodo, corrisponde al terminale di maggior lunghezza, però se per qualche motivo, i terminali vengono tagliati della stessa lunghezza, è possibile distinguere la polarità grazie al contenitore, che ha una parte piatta più vicina al terminale identificato come il catodo.

Come è logico supporre, a maggior resistenza minor illuminazione, quindi minor consumo. In ambienti poco illuminati è sufficiente 1 mA per poter ottenere un buon livello di luce sul LED. I LED forniti sono ad alta efficienza anche se il loro aspetto esterno è lo stesso dei primi LED, che avevano bisogno dai 10 ai 20 mA per ottenere la stessa illuminazione.

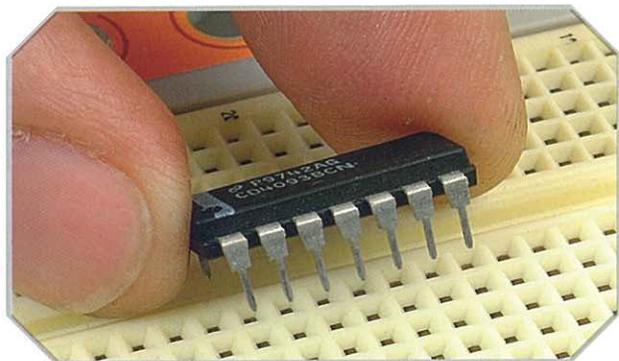
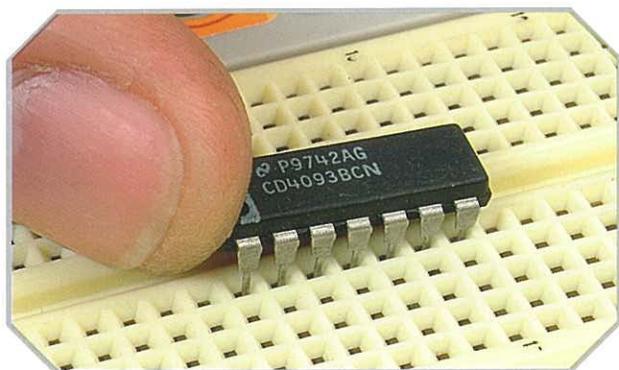
Montaggio degli integrati

La scheda Bread Board permette di recuperare i componenti, quindi sarà necessario utilizzarli con attenzione per poterli riutilizzare il maggior numero di volte possibile. Non è necessario un grande sforzo, basta leggere i suggerimenti che vi indichiamo di seguito.

I terminali dei circuiti integrati devono essere ben diritti, allineati e posti perfettamente davanti ai fori della scheda. Bisogna essere sicuri di questo e, se necessario, "pettinare" i piedini appoggiandoli su una superficie piana e dura, dato che normalmente i terminali vengono forniti più aperti di quanto servirebbe.

Per assicurare un buon contatto elettrico e un buon fissaggio meccanico bisogna inserire l'integrato fino a quando appoggerà sulla scheda, evitando di piegare i terminali.

Smontaggio degli integrati

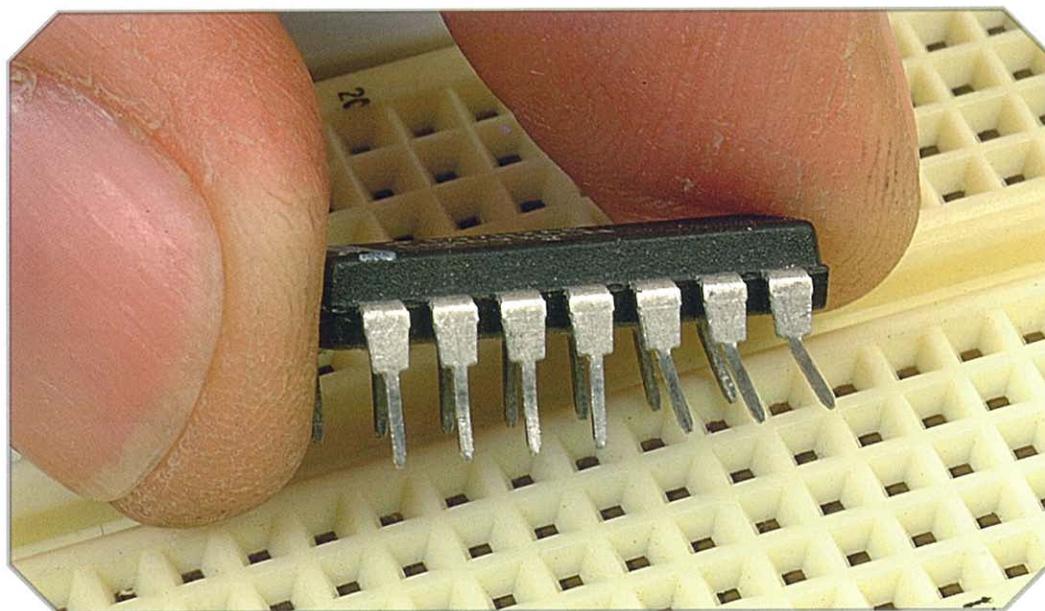
Quando si cambia esperimento, normalmente, è necessario togliere l'integrato; si tratta di un lavoro facile ma bisogna effetturarlo con attenzione. Prima di smontare qualsiasi componente bisogna scollegare l'alimentazione.

Se l'estrazione avviene in modo corretto, i terminali non si danneggiano.

Non bisogna togliere direttamente con la mano l'integrato tirando da un solo lato.

Non si può tirare da un solo lato l'integrato con la mano, perché il componente si inclinerebbe piegando i terminali dell'altro lato, inoltre c'è il rischio di pungersi le dita con i terminali stessi.

L'integrato si toglie molto facilmente se si estrae prima da un lato e dopo dall'altro avvalendosi di un cacciavite, ripetendo l'operazione poco a poco da entrambi i lati, nel caso in cui il componente sia stato inserito con forza. In questo modo l'integrato si estrae e si può togliere con le mani senza sforzo. Tuttavia, non è conveniente toccare i terminali, per cui verrà preso dai due lati. Con questa semplice operazione il circuito integrato può essere inserito e tolto molte volte senza subire danni.

Conservazione

Vi consigliamo di conservare in una piccola scatola tutti i componenti che per qualche ragione non verranno inseriti in modo definitivo nel laboratorio.

I terminali dei circuiti integrati non devono essere toccati con le mani, perché sono sensibili alle scariche elettrostatiche. Questi terminali sono piuttosto delicati e si conservano molto bene se inseriti in un pezzo di polistirolo.

Questo è il risultato per aver cercato di tirare con forza solamente da un lato, l'integrato si inclina e i suoi terminali si piegano.

Astabili con il 4093

In questo esperimento o, per meglio dire, gruppo di esperimenti, si realizzano generatori a onda quadra con il circuito integrato CMOS 4093. Il funzionamento del circuito si può verificare facendo in modo che la frequenza di questi generatori sia tale da poter essere percepita dall'occhio umano, in modo da poter utilizzare un diodo LED come indicatore di funzionamento.

Il 4093

La prima parte dell'esperimento consiste nel verificare il funzionamento del circuito integrato. Basta ricordare la tabella della verità di una porta NAND: per fare in modo che l'uscita sia zero, entrambi gli ingressi devono essere a livello 1, per le altre tre possibili combinazioni l'uscita è 1.

Montaggio

Per questa prima prova è sufficiente inserire il circuito integrato nella zona centrale della scheda Bread Board e collegare un LED su una delle sue uscite, ad esempio sul terminale 10, inserendo una resistenza da 1K8.

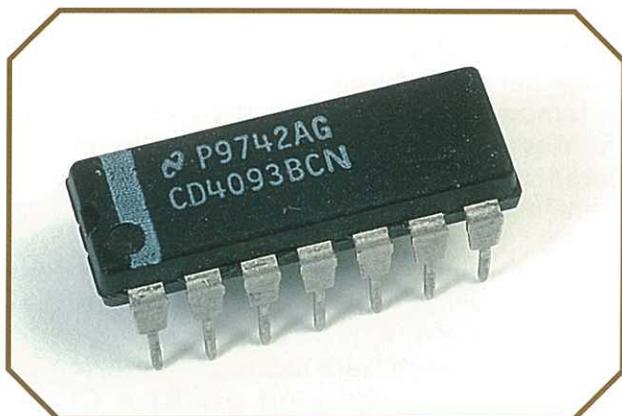
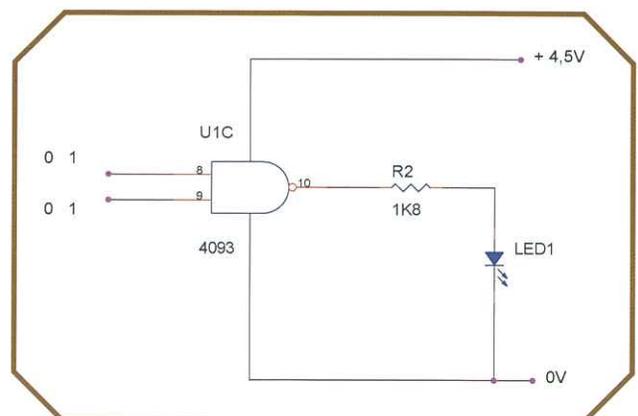
Il catodo del LED deve rimanere sempre verso il positivo dell'alimentazione, la resistenza può essere collegata tra il terminale dell'integrato e l'anodo del LED oppure tra il catodo del LED e il negativo dell'alimentazione. Per quanto riguarda gli ingressi, per ottenere un 1 si collega con un filo al positivo dell'alimentazione, e per ottenere uno 0 si collega al negativo.

Alimentazione

Questo tipo di circuiti CMOS può essere alimentato tra 3,5 e 12 volt, quindi utilizzeremo i 4,5 volt che si possono ottenere facilmente con il portabatterie.

Non dobbiamo dimenticarci di collegare l'alimentazione del circuito integrato: il terminale di alimentazione negativo è sul 7 e quello positivo sul 14.

Esperimento

L'esperimento è molto semplice, basta applicare una combinazione di zeri e uno agli ingressi, terminali 8 e 9, per ottenere l'uscita. Le combinazioni 00, 01 e 10 devono illuminare il LED, mentre con la 11 esso deve rimanere spento. Si può ripetere l'esperimento con ognuna delle porte, perché, anche se può sembrare poco interessante a livello speri-

L'integrato 4093 contiene 4 porte NAND a due ingressi.

Schema di prova del 4093.

Prova del 4093.

Con i due ingressi a 1 il LED non si illumina.

mentale, ci permette di verificare se il circuito integrato funziona o meno, testando il funzionamento di ognuna delle sue porte.

Generatore di onda quadra

Il tipo di porte contenuto nel 4093 ci permette di realizzare degli oscillatori usandone solamente una di esse, sarà possibile anche costruire 4 oscillatori diversi, dato che disponiamo di 4 porte.

All'uscita dell'oscillatore si ottiene un'onda quadra. La frequenza di questo oscillatore dipende dai valori scelti per la resistenza R1 e per il condensatore C1, e si può calcolare in modo approssimativo mediante la formula:

$$f = 1/(0{,}7 \times R1 \times C1).$$

La resistenza R2 ha solamente il compito di limitare la corrente di polarizzazione del diodo LED che si utilizza come indicatore del funzionamento.

Montaggio

Il montaggio è semplice, dobbiamo solamente scegliere una delle porte dell'integrato e collegare le due resistenze, il condensatore e il diodo LED. Prima di collegare l'alimentazione bisogna rivedere tutti i collegamenti; il negativo del condensatore elettrolitico deve essere collegato al negativo dell'alimentazione. L'alimentazione vi è già stata indicata in precedenza ed è stata collegata nell'ultimo esperimento.

Astabile con 4093.

Prove

Vi consigliamo di montare il primo circuito utilizzando i componenti dello schema, ovvero, R1 da 1 M e C1 da 1 µF. La cadenza di illuminazione del LED è bassa, quindi impiega un po' di tempo per accendersi e spegnersi. Per aumentare il tempo potremmo collegare in parallelo con C1 l'altro condensatore da 10 µF, mentre per diminuire il tempo possiamo fare al contrario, cioè abbassare la capacità.

Possiamo anche variare il valore della resistenza R1, e vi consigliamo di provare con i valori di 330 K e 47 K.

È possibile realizzare molte combinazioni di resistenze e condensatori per R1 e C1; ad esempio, con 47 K per R1 e 22 nF per C si ottiene una frequenza elevata, tanto da sembrare

Esperimenti con astabili

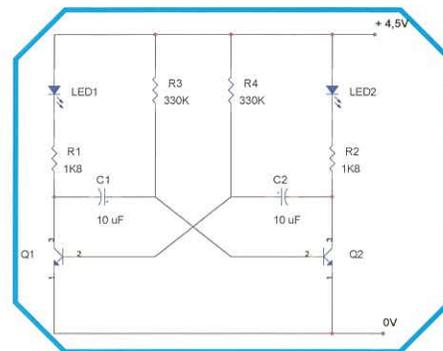
Questo esperimento è basato su una configurazione tipica dell'astabile con transistor e può essere anche utilizzato a frequenze più elevate; abbiamo scelto di utilizzare frequenze molto basse per poter verificare il suo funzionamento mediante l'accensione o lo spegnimento dei due diodi LED utilizzati. Se si utilizzasse una frequenza più veloce, anche se i LED si illuminassero in modo intermittente, li vedremmo sempre accesi a causa della persistenza della retina.

Funzionamento

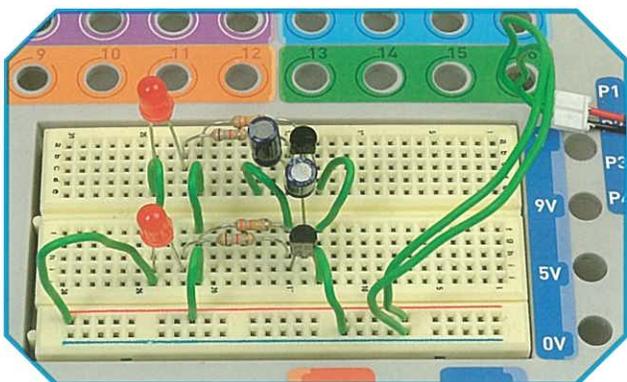
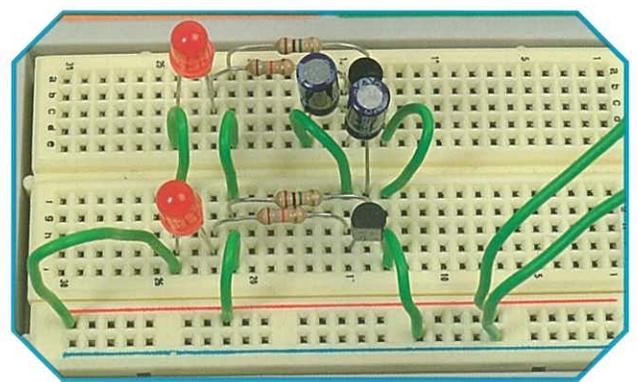
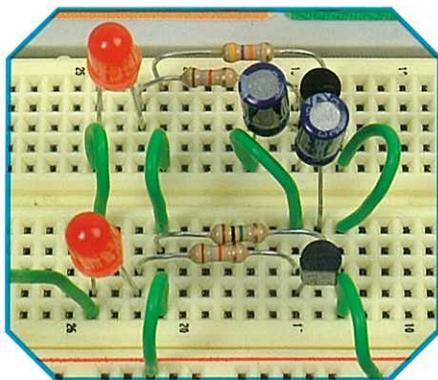
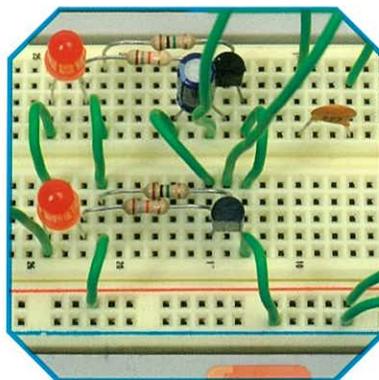
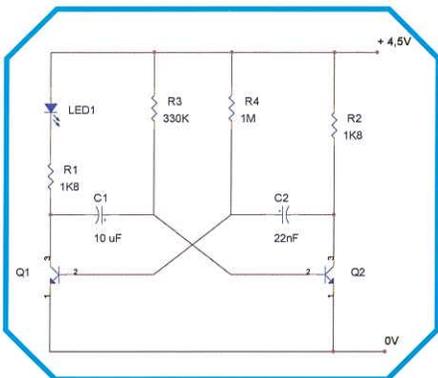
Facciamo una rapida descrizione: i transistor Q1 e Q2 conducono in modo alternativo, ovvero quando uno di essi è in interdizione l'altro è in saturazione, e la polarizzazione di un transistor è controllata dall'altro.

Il condensatore C1 si carica tramite la resistenza R3 in modo che, quando la tensione supera gli 0,6 V sul loro punto di collegamento — questo punto è unito anche alla base di Q2 — quest'ultimo entra in conduzione e passa rapidamente in saturazione, abbassando la tensione sul suo collettore e bloccando la corrente di base del transistor Q1 che cessa di condurre. In questa condizione il LED 2 è illuminato e il LED 1 spento. Questo processo si ripete iniziando, in questo caso, a caricare C2 fino a quando Q1 entra in conduzione, il quale bloccherà Q2 e così via.

Lo schema

Lo schema è piuttosto semplice, dato che ha pochi componenti. Il diodo LED si illumina solamente quando il transistor a cui è collegato conduce, e si spegne quando non conduce. I LED non sono illuminati in modo simultaneo, a questo riguardo, però, occorre fare un'osservazione: se lavorassimo, ad esempio, a una frequenza molto alta, tale che l'occhio umano non la possa percepire, vedremmo i LED costantemente accesi, anche se in realtà si accendono e si spengono alternativamente.

Schema dell'astabile utilizzato in questo esperimento.

Montaggio

Il montaggio di questo esercizio è abbastanza semplice, tenendo conto che i diodi LED hanno polarità e che bisogna fare molta attenzione alla posizione dei transistor. Questo circuito si alimenta con tensione continua compresa tra 3

Ecco come viene realizzato il montaggio dell'astabile.

Aumento del periodo cambiando le resistenze da 1 M a 330 K.

Segnale asimmetrico con una resistenza da 47 K e l'altra da 1 M.

In questo caso un condensatore è da 22 nF e l'altro da 10 µF.

Si possono fare delle prove cambiando i valori dei componenti.

Un condensatore da 44 nF (2 da 22 nF in parallelo).

Il circuito può funzionare con un solo LED, collegando la resistenza da 1K8 direttamente al positivo

e 9 volt, anche se vi consigliamo di utilizzare quella da 4,5 volt. Fino a quando il laboratorio non sarà più completo, potremo utilizzare il portabatterie e tre pile da 1,5 volt.

Esperimento di base

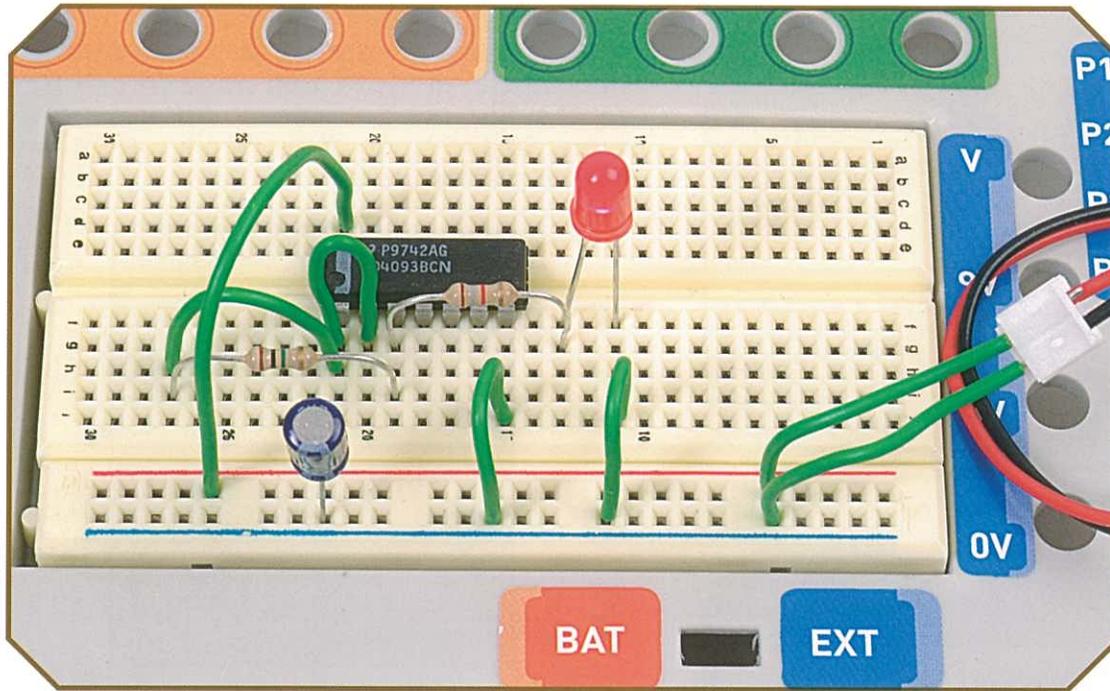
Il circuito, una volta realizzato, deve funzionare fin dal primo momento, tenendo conto che la temporizzazione potrebbe essere piuttosto lunga e che occorrerà aspettare qualche secondo per vederla cambiare. Se non si illumina alcun LED, è necessario scollegare l'alimentazione e rivedere tutti i collegamenti.

Esperimenti

Dopo aver collaudato il circuito si possono fare diversi esperimenti, ottenendo diversi tempi di illuminazione dei LED. Le variazioni devono essere realizzate sulle resistenze R3 e R4 e sulle capacità C1 e C2. Si possono collegare due condensatori in parallelo per ottenere il doppio della capacità, o due resistenze in serie per ottenere il doppio della resistenza. Per R3 e R4 vi consigliamo di provare con 1 M, 330 K, 47 K, e per C1 e C2 con 10 µF, 22 nF e 44 nF. Se i componenti sono dello stesso valore, in modo da rendere il circuito simmetrico, i tempi di illuminazione dei LED saranno uguali, però il circuito deve funzionare anche se non è simmetrico. È anche possibile togliere uno dei LED, però bisogna collegare la resistenza da 1K8 al positivo dell'alimentazione per fare in modo che il circuito funzioni.

LISTA DEI MATERIALI

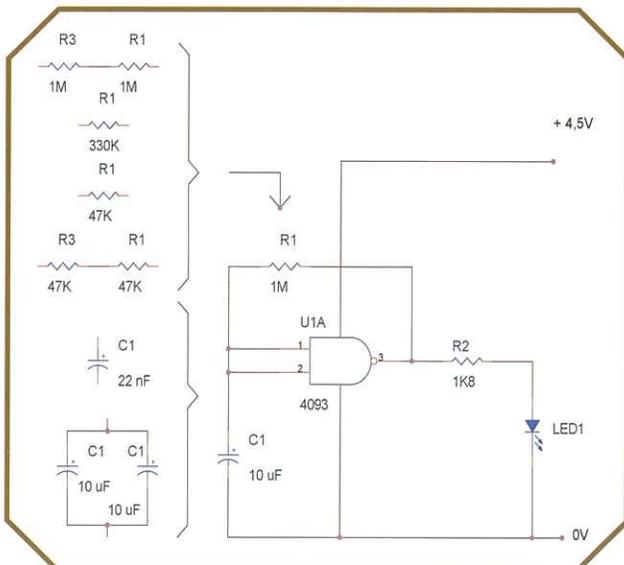
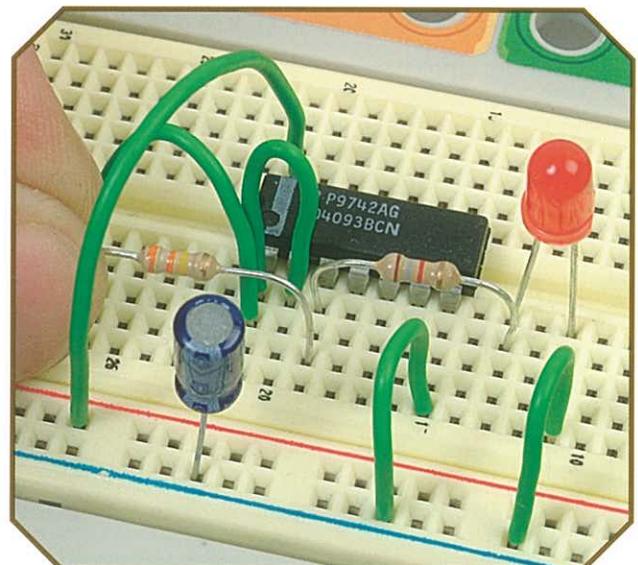
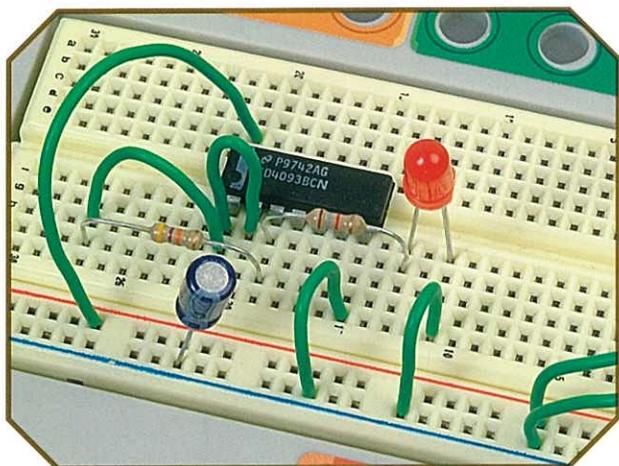
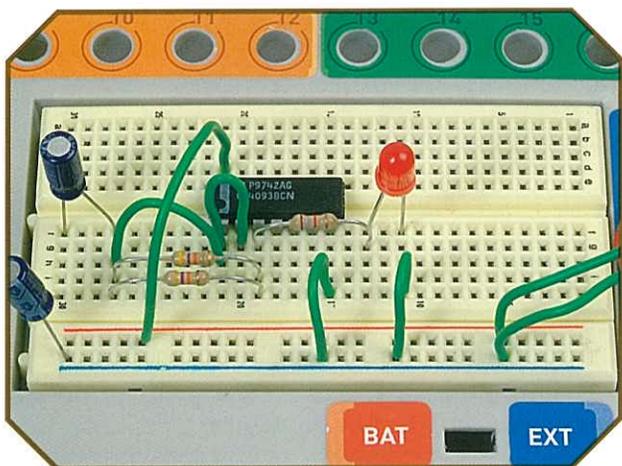
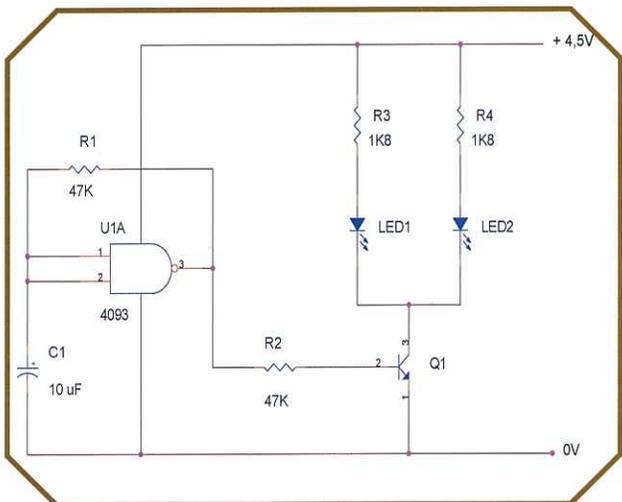
Circuito di base

Q1, Q2	Transistor NPN BC547 o BC548
R1, R2	Resistenza 1K8 (marrone, grigio, rosso)
R2, R3	Resistenza 330 K (arancio, arancio, giallo)
C1,C2	Condensatore 10 µF elettrolitico
LED1, LED2	Diodo LED rosso 5 mm

Generatore astabile con 4093.

che il LED non si spenga, anche se il circuito sta funzionando, malgrado non lo si possa verificare perché l'occhio umano non può percepire variazioni di luce così rapide.

Corrente di uscita

Con i circuiti elettronici è possibile realizzare molti esperimenti, i circuiti logici utilizzati in elettronica digitale utilizzano correnti di uscita e di ingresso normalmente molto ridotte, dal momento che hanno il compito di mantenere un livello di tensione da applicare all'ingresso di un circuito ad alta impedenza, ovvero, che assorbe pochissima corrente.

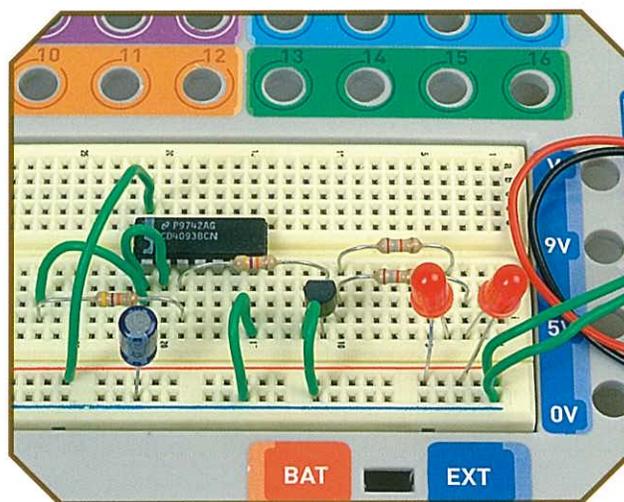
Il 4093 può fornire corrente sufficiente alla sua uscita, anche per alimentare un LED, tuttavia, se non vogliamo sovraccaricare le sue

Schema.

Aumento della frequenza sostituendo R1 con 330 K.

Prova con R1 da 47 K.

Si possono fare diverse combinazioni di componenti, provando anche i collegamenti in serie o in parallelo.

Schema dell'esperimento 3.

uscite o semplicemente se dobbiamo utilizzare un circuito integrato che, pur avendo un buon livello di uscita, non possa fornire corrente sufficiente, potremo modificare il circuito per fare in modo che controlli correnti più grandi alla sua uscita.

LISTA DEI MATERIALI

Circuito di base	
U1	Circuito integrato 4093
R1	Resistenza 1 M (marrone, nero, verde)
R2	Resistenza 1K8 (marrone, grigio, rosso)
C1	Condensatore 10 µF elettrolitico
LED1	Diodo LED rosso 5 mm
Per esperimento 3	
U1	Circuito integrato 4093
R1, R2	Resistenza 47 K (giallo, viola, arancio)
R3, R4	Resistenza 1K8 (marrone, grigio, rosso)
C1	Condensatore 10 µF elettrolitico
Q1	Transistor NPN BC547 o BC548
LED1,LED2	Diodo LED rosso 5 mm

Esperimento 3

Se osserviamo lo schema vedremo un oscillatore astabile costruito con una porta NAND del 4093 e una resistenza da 47 K alla sua uscita che porta direttamente il segnale alla base di un transistor Q1. Quando l'uscita di questo circuito integrato passa a livello alto il transistor conduce producendo l'illuminazione dei LED. Questo transistor può condurre senza problemi 100 mA. In questo caso viene utilizzato per illuminare due LED contemporaneamente.

Con un transistor si controlla la corrente di uscita.

Programmazione: variabili ed espressioni

Ritorniamo alla programmazione spiegando i concetti delle variabili e delle espressioni perché, anche se apparentemente sono concetti semplici, in realtà sono un po' più complessi di ciò che sembra.
Ogni linguaggio di programmazione lavorerà in modo diverso con le variabili e con le espressioni, ma in questo capitolo parleremo del loro utilizzo generale e vi mostreremo la vastità di questi due concetti.

Definizione di variabile

Abbiamo definito una variabile come un deposito contenente un valore che può essere cambiato lungo il corso del programma. Una definizione più tecnica tratterebbe la variabile come un nome per indicare una o più celle di memoria, dove verranno contenuti in modo temporale i diversi valori generati nel programma. In questo modo tali valori potranno essere recuperati quando sarà necessario.

Dichiarazione di una variabile

Quando vogliamo utilizzare una variabile all'interno di un programma normalmente la dobbiamo dichiarare anticipatamente, indicando al processore che vogliamo utilizzare questa variabile nel programma e il tipo di dato che conterrà (a volte è necessario anche inizializzarla).

Per definire una variabile dobbiamo darle un nome e un tipo, e secondo quest'ultimo il compilatore gli riserverà più o meno spazio nella memoria.

Linguaggio	Dichiarazione	Commenti
C	int b	Si dichiara la variabile intera "b"
Visual Basic	Public tempo As Integer	Variabile pubblica "tempo" che sarà di tipo intero
Assembler	PEPE EQU 0A	Si dichiara la variabile "PEPE" che occuperà la posizione 0A della memoria dei dati
PHP	$intero=2004	Si definisce un intero e gli si assegna il valore 2004
FORTRAN 77	Dimensione A (3,3)	Si dichiara un array di 3x3 posizioni

Dichiarazione di variabili nei diversi linguaggi di programmazione.

Esistono linguaggi di programmazione in cui non è necessario definire le variabili. Ad esempio, nel Visual Basic se non si dichiara una variabile il compilatore le assegnerà un tipo per default, con il suo conseguente spazio di memoria.

Anche se non è necessario, è sempre conveniente definire le variabili in precedenza per ottimizzare l'uso della memoria (non utilizzarne più di quella necessaria) e per evitare errori causati dall'incompatibilità dei dati da gestire o dall'aver nominato la variabile in modo inadeguato.

Esistono norme per l'utilizzo delle variabili che devono essere rispettate, perché altrimenti il compilatore darà errore, e dipendono dal linguaggio di programmazione utilizzato. Esisteranno quindi nei diversi linguaggi differenze nella lunghezza del nome della variabile, il tipo di carattere che accettano nel nome, la sensibilità all'utilizzo di maiuscole e minuscole, ecc. In generale, esistono una serie di regole al momento di nominare e utilizzare le variabili e alcune di esse sono consigliabili, mentre altre sono obbligatorie.

Il nome di una variabile deve essere chiaro e descrittivo, deve far riferimento alla sua funzione all'interno del programma
Non possono far parte del nome spazi bianchi o caratteri speciali
Non si devono utilizzare parole riservate del linguaggio stesso per nominare una variabile
Utilizzare sempre maiuscole o sempre minuscole per dare nomi alle variabili, dato che alcuni linguaggi sono sensibili a questa differenza
Non utilizzare all'interno di un programma variabili con lo stesso nome per usi diversi
Indicare la funzione di ogni variabile con un commento
Dichiarare le variabili al livello a cui si utilizzeranno: classe, metodo, procedimento, modulo...

Norme comuni per l'utilizzo delle variabili.

TIPO DI DATO	RANGE 1	DIMENSIONE (byte)
char	-128...127	1
signed char	-128...127	1
unsigned char	0...255	1
byte	0...255	1
int, integer	-32768...32767	2
unsigned int	0...65536	2
signed short	-32768...32767	2
unsigned short	0...65536	2
long	-2147483648...2147483647	4
unsigned long	0...4294967295	4
float	1.40129E-45...3.402823E+38	4
double	4.94065E-324...1.7976931E+308	8
long double	1.2E-4932..	19
boolean	TRUE-FALSE	2
date	Data/ora	8
currency	Valori tipo valuta	
object	Riferimento a un oggetto	4
string	Catene di caratteri	1 per carattere

Classificazione dei tipi di dati.

Tipi di dati

In base ai dati che contengono, le variabili possono essere di molti tipi. Una variabile che contenga i caratteri "CIAO" non può essere considerata uguale a un'altra che contenga il numero "3,14159". Il numero di tipi disponibili dipenderà dal linguaggio di programmazione, e normalmente sono i linguaggi di alto livello quelli che contemplano una maggior gamma di variabili.

Nella tabella della figura è possibile vedere una classificazione dei tipi di dati che possiamo trovare, oltre al range dei valori che possono contenere e alla dimensione della memoria associata a ogni tipo. Come si può supporre, tutte queste caratteristiche variano in base al linguaggio di programmazione.

Oltre ai tipi di dati già visti, alcuni linguaggi possiedono quelli che sono chiamati array o matrici di dati, ovvero insiemi di variabili dei tipi visti, posizionati in modo sequenziale nel-

Errori che rileva il compilatore dichiarando le variabili in linguaggio assembler.

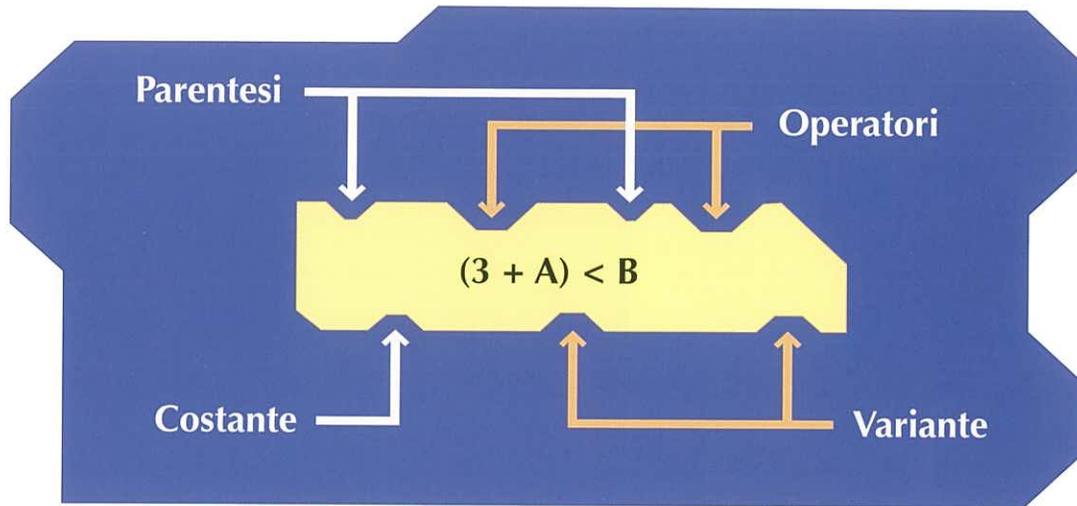
```
typedef struct {
        NOME string (100);
        NOTE int(8);
        RIPETITORE boolean;
}CLASSE;
```

Esempio di una struttura di dati.

la memoria. Quando le variabili di diverso tipo hanno qualcosa in comune sono trattate come un solo dato; alcuni linguaggi permettono la creazione di strutture. Le strutture sono raggruppamenti di variabili in relazione fra loro. Nella figura possiamo vedere un esempio di struttura.

Quando due o più variabili di diverso tipo si trovano nella stessa operazione o espressione, alcuni linguaggi realizzano automaticamente una conversione dei tipi di dati cambiandoli tutti al tipo più restrittivo, invece altri linguaggi in questa situazione generano un errore al momento della compilazione.

Ambito di una variabile

L'ambito di una variabile è la zona del programma in cui questa variabile è visibile e uti-

Esempio di un'espressione e parti che la compongono.

lizzabile. Possiamo quindi fare una distinzione tra variabili globali (pubbliche) o locali (private). Le variabili globali possono essere utilizzate in qualsiasi punto del programma, inclusi tutti i procedimenti e le funzioni, mentre quelle locali vengono utilizzate esclusivamente all'interno del procedimento o funzione in cui sono state create.

Una variabile locale si può creare e inizializzare entrando nel procedimento e viene distrutta all'uscita. Per evitare questo e fare in modo che la variabile si inizializzi solamente la prima volta e poi conservi il suo valore per la chiamata successiva al procedimento, è necessario dichiararla "static".

Operatore	Significato	Tipo di operando	Tipo di risultato
+	Somma	Intero o reale	Intero o reale
-	Sottrazione	Intero o reale	Intero o reale
*, x	Moltiplicazione	Intero o reale	Intero o reale
/, ÷	Divisione	Intero o reale	Reale
**, ^	Esponenziale	Intero o reale	Intero o reale
Div	Quoziente della divisione	Intero o reale	Intero
Mod	Resto della divisione	Intero o reale	Intero
Abs	Valore assoluto	Intero o reale	Intero o reale

Operatori aritmetici più comuni.

a	b	NOT a	a AND b	a OR b
Falso	Falso	Vero	Falso	Falso
Falso	Vero	Vero	Falso	Vero
Vero	Falso	Falso	Falso	Vero
Vero	Vero	Falso	Vero	Vero
		Valore contrario a quello dell'operando	Solo se i due operandi sono veri lo è il risultato	Se uno degli operandi è vero anche il risultato lo è

Tabella della verità del funzionamento dei principali operatori logici.

Definizione di un'espressione

Un'espressione è una composizione ben formata da operandi (variabili e costanti) e operatori (operazioni). Nella figura possiamo vedere un esempio delle parti che possono formare un'espressione semplice: costanti, variabili, operatori, parentesi, ecc.

È necessario conoscere quali simboli sono utilizzati nel linguaggio di programmazione e con quali tipi lavora per poter formare espressioni corrette. Se cercassimo di fare una moltiplicazione scrivendo l'espressione "axb" commetteremmo un errore sintattico, perché l'operatore "x" non è stato implementato nel linguaggio (invece di "x" il linguaggio utilizza "*"). Un errore di tipo sarebbe cercare di sommare "4+False", trattandosi di tipi diversi.

Operatore	Significato
<	Minore di
>	Maggiore di
=, ==	Uguale a
<=	Minore o uguale a
>=	Maggiore o uguale a
<>, !=, /=	Diverso da

Operatori relazionali.

FORTRAN	PASCAL	C
**	*, /, div, mod	Postfisso ++, --
*, /	+, -	Prefisso ++, --
+, -		+, - (unario)
		*, /, %
		+, - (binario)

Regole di precedenza degli operatori aritmetici in funzione del linguaggio di programmazione utilizzato.

Tipi di espressioni

Le espressioni si possono classificare in due grandi gruppi: aritmetiche e logiche. Le operazioni aritmetiche sono simili a quelle matematiche, lavorano con tipi numerici e i simboli che utilizzano sono operatori aritmetici. Si cerca di ottenere un risultato finale che determinerà il valore dell'espressione. Le espressioni logiche, anche dette espressioni booleane, sono composte da costanti e variabili logiche, da operatori logici e da operatori relazionali.

Gli operatori logici più comuni sono "not", "and" e "or". Il primo realizza una negazione del valore su cui opera, il secondo è la congiunzione di due valori (a E b) e l'ultimo la disgiunzione (a O b). Nella figura possiamo vedere come funzionano questi operatori mediante la tabella della verità. In una espressione logica è normale trovare operatori logici insieme a operatori relazionali. Gli operatori relazionali sono mostrati nella tabella allegata.

Valutazione delle espressioni

Per ottenere il risultato di un'espressione è necessario valutarla e, a questo scopo, dobbiamo seguire un determinato ordine. Dobbiamo sapere quali operazioni si realizzano per prime, in quanto l'ordine in cui si eseguono influenza il risultato finale.

A=1, B=2, C=3		
Espressione senza parentesi	Espressione con parentesi	Risultato
A+B*C	A+(B*C)	7
A+B**C/C+A	A+((B**C)/(C+A))	3
A>B AND B<C	(A>B) AND (B<C)	Falso
A+B<=C OR B*A+C<A	((A+B)<=C) OR ((B*(A+C))<A)	Vero
NOT div C/A<B	NOT ((div (C/A))<B)	Falso

Esempio dell'utilizzo degli operatori.

Nelle operazioni aritmetiche l'ordine in cui si valutano gli operandi è definito dalle regole di precedenza e di associatività. La precedenza indica quale operatore si applicherà per primo a fronte dell'esistenza di diversi operatori in un'espressione e l'associatività indica se la valutazione si realizza da sinistra a destra, da destra a sinistra o combinando entrambi i casi. Sia la precedenza che l'associatività dipendono dal linguaggio di programmazione.

In tutti i linguaggi di programmazione le parentesi hanno la massima priorità. Quindi le operazioni interne alle parentesi saranno da valutare per prime. È consigliabile utilizzare le parentesi per evitare qualsiasi dubbio possibile quando utilizziamo un'espressione in un programma.

Nella tabella a sinistra possiamo osservare le regole di precedenza in diversi linguaggi di programmazione.

Negli operatori logici, l'operatore NOT sarebbe il prioritario, seguito da OR e da AND.

Quando si combinano i diversi operatori viene normalmente data la precedenza a quelli aritmetici, poi a quelli relazionali e, in ultimo, a quelli logici.

Le figure allegate riportano diversi esempi di espressioni e dell'utilizzo di operatori al fine di familiarizzare con i concetti appena esposti.